Walkthrough OCCAM

Be on the lookout for this fellow: The call-outs are ACTIONs for you to do!

When you see the check mark, compare your work to the marked element
Objectives

In this presentation you’ll

1. Learn what is OCCAM
   ◦ Why is it needed?
   ◦ What does it do?

2. Learn how to use OCCAM
   ◦ As an experimentalist
   ◦ Using the web interface
   ◦ Running and configuring experiments
OCCAM and the curation of artifacts
Computer Systems

Innovation & Evaluation

- Relies almost exclusively on artifacts
  - Compilers, analyzers, VMs, …
  - Software simulation, …
  - Hardware emulation, …
  - Benchmarks, mini-apps, …
  - Traces, data sets, …
  - And a cast of a thousand other tools…

- *Empirical study* keystone of rapid pace for a huge market (e.g., $111B processor market†)

† IMS Research, 2011
## Prolific Artifact Production

<table>
<thead>
<tr>
<th>Simulators</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>Single-core</td>
</tr>
<tr>
<td>Multi-core</td>
</tr>
<tr>
<td>Homogen. multi-core</td>
</tr>
<tr>
<td>Heterogen. multi-core</td>
</tr>
<tr>
<td>SMT</td>
</tr>
<tr>
<td>Shared memory</td>
</tr>
<tr>
<td>Private memory</td>
</tr>
<tr>
<td>Timer</td>
</tr>
<tr>
<td>Cycle</td>
</tr>
<tr>
<td>Fur</td>
</tr>
<tr>
<td>Full</td>
</tr>
<tr>
<td>Cache</td>
</tr>
<tr>
<td>Cac</td>
</tr>
<tr>
<td>In-chip</td>
</tr>
<tr>
<td>Out-chip</td>
</tr>
<tr>
<td>Sup</td>
</tr>
<tr>
<td>Virt</td>
</tr>
<tr>
<td>Virt</td>
</tr>
<tr>
<td>VLIW</td>
</tr>
<tr>
<td>DRAM controller</td>
</tr>
<tr>
<td>Scheduling</td>
</tr>
<tr>
<td>Concurrency</td>
</tr>
<tr>
<td>DRAM error sim</td>
</tr>
<tr>
<td>On-chip network</td>
</tr>
<tr>
<td>System on chip</td>
</tr>
<tr>
<td>Power consumption</td>
</tr>
<tr>
<td>Gate-level</td>
</tr>
<tr>
<td>Validated</td>
</tr>
</tbody>
</table>

Just a small selection of 31 artifacts!

Tremendously diverse, yet overlapping…
A brewing (brewed?) crisis...

- Symptomatic of fragmented, ad hoc, internal existing effort and investment
- Research expediency and results
  - Can you even find what you need?
  - Duplicating effort to re-implement for comparison
  - May not understand artifact and its use
  - Small incentive to build, release, maintain
  - Creating your own artifacts instead
- Lacks: *accountability* and *leverage*
Community is asking for accountability
- Enabling open simulation & experiments
- Activities by NSF, DOE, DARPA/DoD, CRA, EU Commission, ACM, IEEE, & others over last ten years to inspire, engage the community
- Gov’t. sponsored research – DMP & open access
- ACM is actively studying what to do

- Quality artifacts are available, emerging
- Quality experiments are being done

How can we build and leverage this?
Vision for OCCAM

- Community-supported digital curator & exchange for simulation, emulation, benchmarking, experimental results
- Shared instrument to save time, be fair, advance science: **OCCAM: Open Curation for Computer Architecture Modeling**
- Inspired by Occam's Razor, which suggests minimum assumptions and most succinctness
- Initially *architecture* but compilers & others too.
OCCAM

Building a bridge to accountability

Photo attributed to Sergey Prokudin-Gorsky, http://www.flickr.com/photos/prokudin-gorsky/
OCCAM

Building a bridge to accountability

Photo attributed to Sergey Prokudin-Gorsky, http://www.flickr.com/photos/prokudin-gorsky/
OCCAM

Three pillars

Photo attributed to Sergey Prokudin-Gorsky, http://www.flickr.com/photos/prokudin-gorsky/
Three pillars

Infrastructure Pillar
Three pillars

Infrastructure Pillar

- web portal
- software services
- digital resources
- experiment
- tutorial
- benchmark
- repository
- database servers
- data
- hardware instance (e.g., emulator)
- high-performance cluster
- GPGPU
- FPGA
- emulator on FPGAs
- simulator on CPUs
- simulator on GPGPUs

developer
experimentalist
consumer
manager
Three pillars

Infrastructure Pillar

Holds curated digital objects

**Artifacts**: Simulators, emulators, benchmarks, data sets, etc.

**Experiments**: Methodology, mechanisms, results
Three pillars

Infrastructure Pillar

Operate on digital objects

- Classify, submit, search, retrieve
- Checking with guidelines
- Encapsulated as “runnable” entities
- Visualization, comparative analysis (e.g., param sweeps)
Three pillars

Infrastructure Pillar

- **Infrastructure Pillar**
  - Web portal
  - Software services
  - Repository
  - Digital Resources
  - Hardware Resources

- **Digital Resources**
  - Developer
  - Experimentalist
  - Consumer
  - Repository
  - Artifact
  - Benchmark
  - Tutorial
  - Experiment

- **Hardware Resources**
  - Database servers
  - Simulator on CPUs
  - Simulator on GPGPUs
  - Emulator on FPGAs
  - FPGA
  - GPU
  - Remote hardware instance (e.g., emulator)
  - Remote hardware instance (e.g., high-performance cluster)
Three pillars

Infrastructure Pillar

Hardware resources
Federated, distributed system
• Database engine
• Simulation engine
• Shared: pooled, scarce and/or specialized resources

Remote hardware instance (e.g., emulator)
Remote hardware instance (e.g., high-performance cluster)

Manager

Digital Resources

Hardware Resources
Distributed Repository

Per user (group) instance
Services, repository
Private, secure results
Distributed Repository

New capabilities in OCCAM
Sharing artifacts, and results
Services on shared objects
Access rights & groups
Push/pull distribution
Sharing Resources

OCCAM local instance
Access to specialized resource
Shared among multiple users

New capabilities in OCCAM
Interfaces to the HW/SW
Mediate access (rights, sched.)
Privacy/security of usage
Export/import of results
Sharing Resources

access to scarce/proprietary HW e.g., HMC, FPGA shared to group
Sharing Resources

access to SW tools
e.g., proprietary sim.
ship benchmark to sim.
combination of HW+SW
Sharing Resources

access to SW tools
e.g., proprietary sim.
ship benchmark to sim.
combination of HW+SW
Three pillars

Community Pillar

- **Create and foster a community**
  - Establish procedures and policies
  - Contribute artifacts and experiments
  - Ingrain and use repository, e.g., conference submission and publication of experiments
  - Advocate and convince people of value and values behind open-access repository
Three pillars

Education Pillar

- Education materials & activities to train the community, esp. new students
- Training & materials
  - Lower barrier to adoption
  - Experimental methodologies
  - Best practices for development, experimentation
  - Bootcamps & short training
Three pillars

Oriented Around Value

- Today’s Island model: *Work individually, occasionally share*
  - We’re used to operating this way!
  - Limited packaging, distribution needs
  - Don’t have to support something
  - Less testing, might validate, gotchas & glitches OK, lots of duct tape
  - Lightweight, flexible approach for research
  - Focus put on producing results, research expediency
- Island model is successful (we do innovate).
- Yet, *many challenges & impeding research* with this model!
  - Lack leverage, significant wasted effort, poor scientific method
  - Impedes research progress due to the wasted time, credibility
Changing the Value Balance

The perceived value

Create more value from open-access exchange
Judge decisions in building the exchange on value creation

Today: Easier to “roll your own” but value is recognized in distribution

<table>
<thead>
<tr>
<th>No packaging</th>
<th>Simple test</th>
<th>No support</th>
<th>Deployment</th>
<th>Leverage</th>
<th>Management</th>
<th>Visualization</th>
<th>Correlation</th>
<th>Diagnosis</th>
<th>Analytics</th>
<th>Recognition</th>
<th>Reward</th>
<th>Access</th>
<th>Publishing</th>
<th>Financial</th>
</tr>
</thead>
</table>

Perceived Value
Changing the Value Balance

The perceived value

Create more value from open-access exchange
Judge decisions in building the exchange on value creation

Leveraging other experiments to reduce burden for comparison

Perceived Value
Changing the Value Balance

The perceived value

Create more value from open-access exchange

Judge decisions in building the exchange on value creation

More than simply leverage:
Compelling services for users/developers
• Data management (DOE/NSF)
• Data visualization
• Data correlation/discovery
• Simulation analytics and diagnostics

Perceived Value
Changing the Value Balance

The perceived value

Create more value from open-access exchange

Judge decisions in building the exchange on value creation

Community & education providing tangible incentives
- Recognition & reward for artifacts, experiments
  - e.g., www.artifact-eval.org
- Access to data, services, unique hardware/tools in return
Changing the Value Balance

The perceived value

Create *more value* from open-access exchange

Judge decisions in building the exchange on value creation

Community & education providing *tangible incentives*
- **Publishing**
  - Incorporate as part of reviewing process for journals/conferences
- **Services or financial**
  - Software Institute to help develop, manage, train

---

Perceived Value
Changing the Value Balance

The *perceived* value

Create *more value* from open-access exchange

Judge decisions in building the exchange on value creation

--

**Perceived Value**

- No packaging
- Simple test
- No support
- Deployment
- Leverage
- Management
- Visualization
- Correlation
- Diagnosis
- Analytics
- Recognition
- Reward
- Access
- Publishing
- Financial
OCCAM pillars

Starting point

Where does OCCAM stand?
Starting Point

- **Infrastructure Pillar**
  - End-to-end pass to conduct, share experiments

- **Community Pillar**
  - Engage and build a community
  - Raise awareness, attract contribution

- **Education Pillar**
  - Experimental methodology for architecture
  - Share training and other best practices
    - OCCAM is being used in classes at U. Pittsburgh
Using OCCAM – The experimentalist’s perspective
Step 1: Get a simulator

Obtain simulator

request

metadata

OCCAM web portal
Step 2: Import to Experiment

OCCAM web portal

build & check
Step 2: Import to Experiment

OCCAM web portal

build & check

provenance

sim build

src + vm

bin + env
Step 3: Define/Run Experiment

Experiment Editor
Configuration Editor
Form, visual, import

define & run experiment

OCCAM web portal

build & check
config instantiate dataset
exp. 1

sim build

provenance
Step 3: Define/Run Experiment

Define & run experiment

OCCAM web portal

build & check

cfg

inst

stantiate
dataset

Pulls simulator + environment
Schedules & monitors simulations
Extracts results upon completion
Everything run in its own separate VM
Multiple job types (build, simrun, etc.)
Step 3: Define/Run Experiment

OCCAM web portal

define & run experiment

build & check
config instantiate dataset

exp.1

results

provenance

sim build
Step 4: View Results

OCCAM web portal

build & check
config instantiate dataset
view results
results

create graph

Or data

provenance
sim build
Containerization

- OCCAM runs simulators with Docker
  - Lightweight “virtualization”
  - Local & in-repository build & run

- Docker is used to create running environments
  - A Docker base image
  - Extended with dependences
  - Docker specification generated
What we are doing...

Learn how to use OCCAM as an experimentalist

1. **Import** a simulator into OCCAM
2. **Create** and **configure** an experiment
   - Experiment: Metadata to run a simulation

Due to time constraints, a Docker container with SST is already on your system
What we are doing...

Learn how to use OCCAM as an experimentalist

1. **Import** a simulator into OCCAM
2. **Create** and **configure** an experiment
   - Experiment: Metadata to run a simulation
3. **Launch** (run) experiment
   - Schedules the simulation on a simulation engine
4. **Visualise** experimental results
   - Interactive plots
OCCAM user interfaces

- OCCAM has two means of interaction
  - CLI – Command Line Interface
    - Advanced users (developers)
    - Using scripts
  - WI – Web Interface (Graphical)
    - For everyone
    - Create, configure, and run experiments

We will be using this for now
Accessing the web interface

Use your laptop’s web browser to access your OCCAM instance

- Follow instructions in the handout

(a) Access to https://<IP ADDRESS>

Warning: Your browser may complain about the certificate. When this happens, follow the browser instructions to add an exception.
Home page
Create a user

(a) Click Signup

(b) Type a username & password

(c) Click “Sign Up”
Creating a Workset

- Worksets are like folders
  - They hold the different OCCAM objects
    - Simulators, experiments, etc.

(a) Type a name for a new workset e.g.: “Exercise 1”
(b) Click to create
Workset

Workset settings

Contributors

Content

Authors

Collaborators

Directory

To begin, you'll want to create an Experiment. An Experiment allows you to select and configure tools to run on a machine. Select "experiment" and type in a name to identify it and click "add" and it will take you to a page where you can start playing.

Import a New Object

Add a New Object
Workset

Public/Private

To begin, you’ll want to create an Experiment. An Experiment allows you to select and configure tools to run on a machine. Select "experiment" and type in a name to identify it and click "add" and it will take you to a page where you can start playing.

Import a New Object

Add a New Object
Add a bookmark (user screen)

Directory

To begin, you’ll want to create an Experiment. An Experiment allows you to select and configure tools to run on a machine. Select "experiment" and type in a name to identify it and click “add” and it will take you to a page where you can start playing.

Import a New Object

Add a New Object
Workset

Clone the workset

Authors
new_user
username

Directory
To begin, you’ll want to create an Experiment. An Experiment allows you to select and configure tools to run on a machine. Select "experiment" and type in a name to identify it and click "add" and it will take you to a page where you can start playing.

Import a New Object

Add a New Object

54
Workset

View workset history

To begin, you'll want to create an Experiment. An Experiment allows you to select and configure tools to run on a machine. Select "experiment" and type in a name to identify it and click "add" and it will take you to a page where you can start playing.

Import a New Object

Add a New Object
Workset

Authors Have permissions to modify all contents of a Workset.

Collaborators Can be given have different permissions than authors.
Workset

Import objects (e.g. github)

Create new objects (e.g. experiments)
Importing OCCAM objects into a Workset

- Can be imported from a git repository
  - Lets import an object from bitbucket
  - Find the link in your handouts
Importing existing objects

You’ve just imported a simulator

- Simple MIPS processor
  - Lets use it in an experiment

You’ll see the imported simulator here
Creating a new experiment

You’ve just imported a simulator

- Simple MIPS processor using the SST framework
  - Lets use it in an experiment
- Lets create an experiment that
  - Runs the simulator you imported
  - Plots the results for inspection
Experiments – Workflows

First create a new experiment

(a) Select type experiment
(b) Give it a name
(c) Click add
Experiments

Workflow

A workflow describes how a collection of objects interact. Generally, it denotes a path from some input to some eventual result. For instance, you can describe a flow from a benchmark to a simulator. To do this, you would first attach a simulator. At that point, it will show you the possible input objects you can attach. You can select a benchmark on the left-hand side and all runs will use that benchmark with the given simulator.

Configurations

There are no workflow nodes. Please add one above to see a list of configuration options for that tool.
Experiments

Workflow

Workflows are DAGs that represent the execution of an experiment

- This one is empty, let's add the plotting tool
- Experiments are built from the end

A workflow describes how a collection of objects interact. Generally, it denotes a path from some input to some eventual result. For instance, you can describe a flow from a benchmark to a simulator. To do this, you would first attach a simulator. At that point, it will show you the possible input objects you can attach. You can select a benchmark on the left-hand side and all runs will use that benchmark with the given simulator.
(a) Click the empty node
(b) Enter the type: script
(c) Enter the name: plotter
(d) Click attach
The plotting tool is in the workflow!

The final block on the workflow represents the output of the experiment.
Experiments

Workflow

This plotting tool is can plot the output of the simulator

- The next step is to add the inputs to the plot
- We’ll be adding the simulator we imported
- However we need to add an intermediate block
  - An intermediate storage for the simulator output
Experiments

Workflow

(a) Click the plus sign
(b) Enter the type: application/json
(c) Select the option: {new application/json}
(d) Click attach
Experiments

Workflow

(a) Click the plus sign

(b) Enter the type: simulator

(c) Enter the name: XSim-demo

(d) Click attach
Experiments

Workflow

The type of the intermediate block is the same as the output of our simulator and the input of the plotting tool.
Experiments

Workflow

Finally let's add some input to the simulator

- We've provided a program to you

(a) Click the plus sign

(b) Enter the type: program/XSim

(c) Enter the name: XSim-program

(d) Click attach
Experiments

Workflow

We’ve just created a workflow.

Now we can configure it.
Experiments

Configurations

Configuration options

Xsim-demo configurations

Name of experiment: Xsim-results

XSim

The frequency of the CPU: 2MHz

Memory options

Memory Size: 2MiB

Clock frequency (Can be a comma separated list of frequencies): 200Hz, 2KHz, 20KHz, 200KHz

Backend:

SimpleMem

Verbose: 0

Access time: 100ns
Experiments

Configurations

(a) Change the Xsim-demo configurations as you see fit

(b) Click update to save any changes
Experiments

Configurations

(a) These options are configured by default.

(b) DO NOT CHANGE THESE
These options select the data to be plotted from the simulation output.
Experiments

Configurations

(a) You can also try to change the plotter - Figure Options

(b) Click update to save any changes
Running an experiment is as easy as clicking a button!

(a) Click on the run tab
Running an experiment is as easy as clicking a button! (or two 😊)

(b) Click run
Experiments

Running

If you see some errors like these, don’t worry! Docker is just a bit verbose when checking if the containers exist, and when they do not OCCAM creates them!

```
Error response from daemon: No such container: occam-clbc45fe-5545-11e7-
acdc-2c4d5405fa17-2827d24833a0417933508f8bbfec475c692a0ff3
Error response from daemon: No such image: occam/clbc45fe-5545-11e7-
acdc-2c4d5405fa17-2827d24833a0417933508f8bbfec475c692a0ff3
Sending build context to Docker daemon 61.44 KB
Step 1/5 : FROM occam/0ab7be68-0514-11e7-93ce-1c1b0d00a9044:f56c167c2515ad1f9bcb297ebba0cde31cc2978
         ---> 745596892d74
Step 2/5 : ADD . /occam/clbc45fe-5545-11e7-
acdc-2c4d5405fa17-2827d24833a0417933508f8bbfec475c692a0ff3
         ---> c960834c5399
Removing intermediate container cd0f3c1f4cf5
Step 3/5 : RUN /bin/bash -l -c cd /occam/clbc45fe-5545-11e7-
acdc-2c4d5405fa17-2827d24833a0417933508f8bbfec475c692a0ff3; /bin/true
         ---> Running in 3ab768a38a75
[91mbash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
[km:91m:91m:91m: Inappropriate ioctl for device
/km:91m:91m:91m: No such file or directory
/km:91m:91m:91m: Previous command repeated
/km:91m:91m:91m: Previous command repeated
```
Experiments

Running

(a) Once your run is complete, you need to refresh the page to see the results.
Experiments

Running

(a) Click the Output tab

(b) Open the plot
Experiments

Running

(a) Your plot should look like this

(b) Check the provenance tab of the plot
Experiments

Running

(a) You can check how this plot was created

The software that created it

Inspect the experiment that generated it

The machine where it ran
Summary

We presented OCCAM

- A community-supported digital curator & exchange for simulation, emulation, benchmarking, experimental results

- OCCAM allows to easily deploy and run software
  - Making reproducing results easy
  - An fostering software reutilization

- OCCAM is easy to use
  - But how easy is it to create a simulation in OCCAM?
  - We’ll create a simulation in the next exercise
Acknowledgments

We gratefully acknowledge the support and collaboration of our research partners.

Portions of this material is based in part upon work supported by the National Science Foundation under Grant Numbers CCF-142331, CNS-1012070, ACI-1535232, CNS-1305220 and CCF-1148646. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.